National Institutes of Health

Division of Cancer Prevention

NASA

California Institute of Technology 1891

National Cancer Institute

Fred Hutchinson Cancer Research Center
A LIFE OF SCIENCE

JPL

# EDRN Software Products

## Release, Installation, and Maintenance Guide

*NASA Jet Propulsion Laboratory* 4800 Oak Grove Drive Pasadena CA 91109-8001 http://www.jpl.nasa.gov/

# Signature Page

This is an *official* document representing the policies, recommendations, technical considerations, and other aspects of the Early Detection Research Network's software products as codeveloped by EDRN's Data Management and Coordinating Center and NASA's Jet Propulsion Laboratory in its role as Informatics Center. As such, the following signatures signify *acceptance* and *endorsement* of this document.

______________________________

Sudhir Srivastava, Ph.D., M.P.H., Chief, Cancer Biomarkers Research Group

______________________________

Ziding Feng, Ph.D., EDRN Principal Investigator, DMCC

______________________________

Daniel Crichton, M.S., EDRN Principal Investigator

# Table of Contents

**Introduction** 5

**How this Document is Organized** 6

**Intended Audience** 6

**Requirements** 7

**Background** 7

Developing Informatics Applications 7

Providing a Public Image 8

Supporting Science 8

Maturing the Model 8

**Overall Requirements** 9

Functional Requirements 9

Nonfunctional Requirements 9

Application-Specific Requirements 10

Interoperability 10

**System Architecture** 11

Component Overview 11

Deployment Model 13

**Release Engineering** 13

Identifying Releases 14

**Public Portal** 15

**Requirements** 16

**Component Specifications** 17

Hardware Platform 18

Software Specifications 19

*Installing the Prerequisite Software* 19

*Testing the Prerequisite Software* 22

**Installing the Public Portal Products** 23

Installing the Products 24

Testing the Installation 24

*Checking the Portal Skin* 24

*Checking the ERNE Application* 25
*Checking the EDRN Knowledge Environment* 25
**Regular Maintenance** 26
Backups 26
**Study Information Management** 29
**Catalog and Archive Service** 31
**Biomarker Database** 33
**Virtual Specimen Bank** 35
**Secure Site** 37
**Appendix** 39

# Introduction

The Early Detection Research Network (EDRN) researches and develops biomarkers and technologies for the clinical application of early cancer detection strategies. EDRN is an initiative of the National Cancer Institute (NCI) and brings together dozens of institutions to help accelerate the translation of biomarker information into clinical applications and to evaluate new ways of testing cancer in its earliest stages and for cancer risk.

Since its founding in April 2000▸, EDRN has fostered a highly collaborative, multiple disciplinary research venue to improve its techniques and methods. *Informatics* has been a core element of this collaboration. Computing technology and dense data dissemination have enabled the sharing, discovery, correlation, and synthesis of cancer research knowledge in efficient and often novel ways.

*The Early Detection Research Network* Third Report, March 2005.

Two components of EDRN, the Data Management and Coordinating Center (DMCC) run by Fred Hutchinson Cancer Research Center and the Informatics Center (IC) run by NASA's Jet Propulsion Laboratory (operated by the California Institute of Technology) have been at the forefront of EDRN's informatics efforts. Together, the DMCC and the IC have developed and deployed *web-based applications* that facilitate and foster the goals of the EDRN. Throughout all the phases of EDRN's efforts, from preclinical exploratory, to clinical assay and validation, to retrospective longitudinal, to prospective screening, and to cancer control, EDRN's informatics played pivotal roles.

Now, in 2006, EDRN plans to launch several pioneering new applications. However, in order to make such launches successful, EDRN recognizes the timeliness of establishing *rigorous processes and metrics* for software development, deployment, testing, and maintenance. The previous ad hoc processes can no longer support nationwide software deployments in any reasonable fashion. After all, the quality of a system is influenced by the quality of the process used to acquire, develop, and maintain it▸.

*Capability Maturity Model Integration (CMMI) Overview*, Carnegie Mellon Software Engineering Institute, 2006.

Therefore, this document serves to establish those processes and metrics necessary for undertakings of this scale, to provide testing procedures, and to demonstrate how to install and maintain such software.

## How this Document is Organized

This document is grouped in major sections to first give overall requirements and system architecture, and then by major sections for each EDRN application. The following list gives a summary of the major sections:

- *Requirements* gives background of the informatics projects and grounds them as well as future applications in a set of functional (and other) requirements. It also delineates the overall system architecture, providing context for each of the EDRN applications and their relationships.
- *Public Portal* describes installation, operations, and maintenance of the public facing EDRN website that enables EDRN members, non-members, and the general public to learn about EDRN's efforts and utilize various web-based applications.
- *Catalog and Archive Service* describes *eCAS*, the distributed system for saving and indexing scientific data at any stage in the collection pipeline.
- *Biomarker Database* tells how to install, operate, and maintain the centralized database of biomarkers that enables researchers to track the progress of biomarker development.
- *Virtual Specimen Bank* gives installation, maintenance, and other advice on the EDRN Resource Network Exchange, ERNE, the virtual specimen bank application.
- *Secure Site* tells how to install and maintain the members-only EDRN web site.
- *Study Management* tells how to operate and maintain the Virtual Study Information Management System, VSIMS.

## Intended Audience

Those who are tasked with developing or maintaining any of the EDRN applications will want to understand the *Requirements* section as well as the specific sections dealing with the applications of interest. For example, a researcher intending to install an eCAS system will want to read both the *Requirements* and *Catalog and Archive Service* sections.

# Requirements

In this section, we'll give background on EDRN's informatics projects, discuss the general requirements for EDRN's current and future applications, and look at the overall system architecture out of which we can use interoperating applications to construct an entire *system architecture.*

## Background

When the Early Detection Implementation Group proposed the concept for the Early Detection Research Network, one of the constituent parts of the program was a Data Management and Coordinating Center (DMCC)▸. The DMCC provides statistical, computational, and logistical, support to EDRN including study design consultation, data analysis, and meeting coordination. In addition, the DMCC develops web applications to serve EDRN.

*The Early Detection Research Network*, Initial Report, October 2000.

In 2002, the National Cancer Institute (NCI) entered into an interagency agreement with the National Aeronautics and Space Administration (NASA), specifically NASA's Jet Propulsion Laboratory (JPL), operated by the California Institute of Technology. Using software developed by JPL to manage astrophysical, planetary, and astronomical science, EDRN worked with JPL to deploy a next-generation collaborative system called the EDRN Resource Network Exchange (ERNE)▸. ERNE created a virtual specimen bank by dynamically unifying existing specimen repositories at participating EDRN sites. It was to be the first in a series of joint efforts towards improving and pioneering informatics.

*The Early Detection Research Network*, Second Report, October 2002.

Since then, EDRN has declared JPL its *Informatics Center*, supporting the DMCC and the rest of EDRN in its informatics efforts. Indeed, this very document was authored by the Informatics Center.

### Developing Informatics Applications

The relationship that JPL has provided to EDRN has been fruitful, and indeed, the Informatics Center has developed and assisted in the development of additional informatics applications.

While the DMCC began development of the Virtual Study Information Management System (VSIMS), the Informatics Center retrofitted an existing artifact tracking system▸ to serve as the issue tracker for VSIMS. The Informatics Center also assisted with the development of the EDRN Common Data Elements (CDEs) and with bringing additional ERNE sites online.

http://scarab.tigris.org/, which the Informatics Center can no longer recommend.

## Providing a Public Image

EDRN turned to the Informatics Center to architect and deploy the EDRN Public Portal, a public website▸ that enables all classes of users, from EDRN member scientists, to potential collaborators, to cancer patients and advocates, and the curious public to learn more about EDRN, its various programs, funding opportunities, publications, tools, and more. With timely information housed by the DMCC for EDRN's members-only website, the triple effort of DMCC, National Cancer Institute, and Informatics Center made the public portal a reality.

http://edrn.nci.nih.gov/

Through the vehicle of the public portal, the Informatics Center hopes to deploy a pioneering set of new informatics applications.

## Supporting Science

Drawing upon its experience with cataloging and archiving data from space missions, the Informatics Center at JPL began architecting and developing a new generation of applications to support cancer research within the EDRN. These developments include the EDRN Catalog and Archive Service (eCAS) and the Biomarker Database. In parallel, the DMCC has been perfecting VSIMS and developing additional applications.

Unifying these disparate applications are the EDRN CDEs. Through the CDEs, applications interchange data with a single, interoperable vocabulary. This prevents applications from becoming defunct with knowledge stored in inaccessible silos. Moreover, automated correlation and synthesis of knowledge is possible, as is unified and cross-discipline search and discovery.

## Maturing the Model

The ad hoc nature of both the DMCC's and Informatics Center's software releases served EDRN well enough during its formative stages. When, for example, the DMCC must take down all of VSIMS in order to integrate a new study, such unavailability is usually tolerable.

However, with the imminent "first customer" releases of important new applications (eCAS and Biomarker Database), and the first major injection of applications into the EDRN Public Portal (a unified knowledge environment), the Informatics Center felt it was overdue in providing a mature set of release, installation, operation, and maintenance processes that would ensure the quality and availability of informatics software.

This document serves to capture the release engineering processes, installation processes, maintenance processes, and other related information on EDRN's software products.

# Overall Requirements

This section presents, in no particular order, the requirements that apply in general to past, current, and future EDRN informatics applications.

## Functional Requirements

Functional requirements are requirements that constrain as well as describe the features of EDRN informatics applications. The following list presents these requirements:

- Applications shall annotate all output data with the EDRN Common Data Elements (CDEs). By marking data with appropriate CDE metadata▸, applications ensure interoperability.

Metadata, or data *about* data, is vital for interoperability. With metadata, two applications can interpret a value like "37" as degrees centigrade of body temperature without ambiguity.

- Applications shall accept input annotated with the EDRN CDEs. The CDEs provide applications with an automated means of handling input. Where appropriate, applications may discard data not relevant to its function.
- Applications shall store persistent data annotated with the EDRN CDEs. By persisting data with CDE metadata, application developers can hope to recover knowledge in the event of catastrophic failures.
- Applications shall leverage asymmetric encryption for intercommunication, where appropriate. As these applications may deal with human subjects or other personally identifying data, encryption is a must. Encryption key length may be determined on a per-application basis.
- Applications shall authenticate and authorize users for access where such authentication and authorization are necessary.
- Applications shall leverage industry standards (such as TCP/IP, TLS, XML, and RDF) where appropriate.

## Nonfunctional Requirements

Nonfunctional requirements constrain and describe those parts of applications not directly related to their functioning operation. The following list presents these requirements:

- Applications shall provide user interfaces (where appropriate) using web technologies. Web-based (browser, HTML, and HTTP) user interfaces automatically and transparently provide for remote control and a familiar feel for today's users.

- Applications shall respond to users' requests rapidly. While we recognize that various application features and user demands will vary, we also acknowledge that users are far more pleased and more likely to re-use and recommend applications that feel zippy and quick. For web-based applications, a single second response time is optimal, while 10 seconds are the utmost upper limit▸.

Jakob Nielsen, *Usability Engineering*, Morgan Kaufman Publishers, 1994.

- Desktop applications shall be portable where possible. Users' preferences in hardware and operating system platforms vary greatly, and it is unreasonable to specify a reference platform for desktop software. As such, portability across platforms is a necessity. (For server-based software, we recognize that reference systems may be specified for certain performance and/or maintenance requirements.)
- Server-based applications shall be available at least 12 hours a day, 5 days a week (Monday through Friday).
- Applications shall provide single-click or wizard-based installation that poses a minimum number of questions upon users. Applications shall include supporting software where target platforms do not provide such software (for example, neither Python nor Java are standard on Windows systems, thus Windows-based installers must include them; both are available on Mac OS X systems, so such installers may omit them). This is an absolute requirement for desktop software; it is encouraged, but optional, for server-based software.
- Applications must provide documentation, such as (but not limited to) online help, hypertext, reference manuals, tutorials, user guides, and printed matter.
- Applications shall support distributed, federated, and centralized deployments as necessary.
- Applications shall adhere to Federal and state regulations including but not limited to Institutional Review Board (IRB), Department of Health and Human Services (DHHS), Federal Drug Administration (FDA), and Health Insurance Portability and Accountability Act (HIPAA).

## Application-Specific Requirements

Requirements for each EDRN informatics application are detailed within their sections.

## Interoperability

A core requirement of every application within the entire knowledge environment is interoperability. As stated above in the functional requirements all output data and all input data must be annotated with the common data elements. This is a pivotal feature for interoperability and sharing of data between applications.

To annotate data with metadata, applications must leverage a standard format that uses the Common Data Elements. We therefore stipulate that the Resource Description Format (RDF)▸ be the technology to annotate data with metadata. RDF is a widely recognized standard for metadata and is fully compatible with the EDRN CDEs.

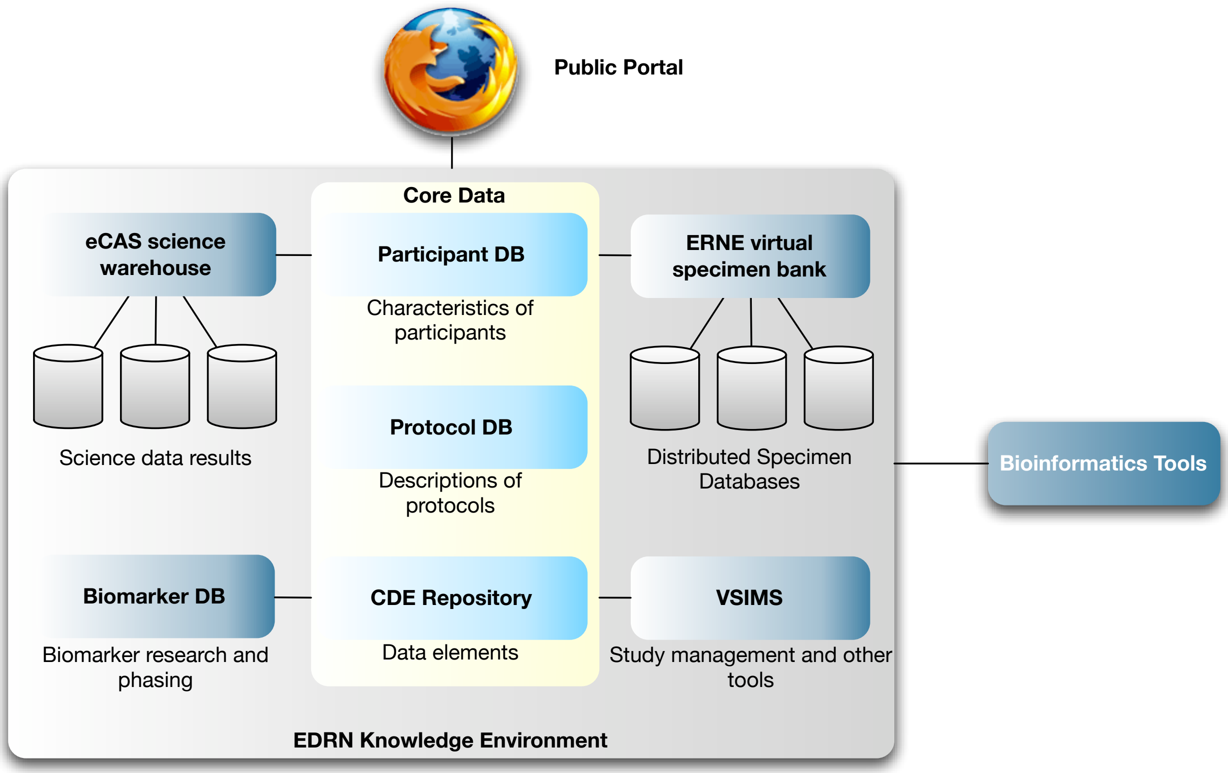
http://www.w3.org/RDF/

# System Architecture

To support these general requirements, and to provide an infrastructure into which EDRN informatics applications—both present and future—can exist, the Informatics Center along with the DMCC have defined an overarching architecture consisting of both hardware and software platforms. This section presents multiple views of the architecture in order to make it visible and understandable to the reader.

## Component Overview

Several system components comprise the complete set of EDRN informatics applications. We call this set the EDRN Knowledge Environment (EKE). The EKE consists of separate yet interoperating applications that provide specific needs while enabling cross-correlation between study disciplines. The following diagram depicts the EKE:


Public Portal
Core Data
eCAS science warehouse
Participant DB
Characteristics of participants
ERNE virtual specimen bank
Science data results
Protocol DB
Descriptions of protocols
Distributed Specimen Databases
Bioinformatics Tools
Biomarker DB
Biomarker research and phasing
CDE Repository
Data elements
VSIMS
Study management and other tools
EDRN Knowledge Environment

It consists of the following components:

- *Core data* is maintained by the DMCC and tracks participants, protocols, and other data (not shown). It also houses the CDEs and serves as the definitive source for CDE definitions. The DMCC must provide an RDFS interface to the CDEs.

- *eCAS* installation may be installed wherever appropriate (personal, workgroup, institution, all of EDRN).

- A single *Biomarker Database* tracks all biomarker research.

- *ERNE* installations at participating sites handle queries for specimen data. A single ERNE web application serves as the user interface to the entire network of ERNE sites.

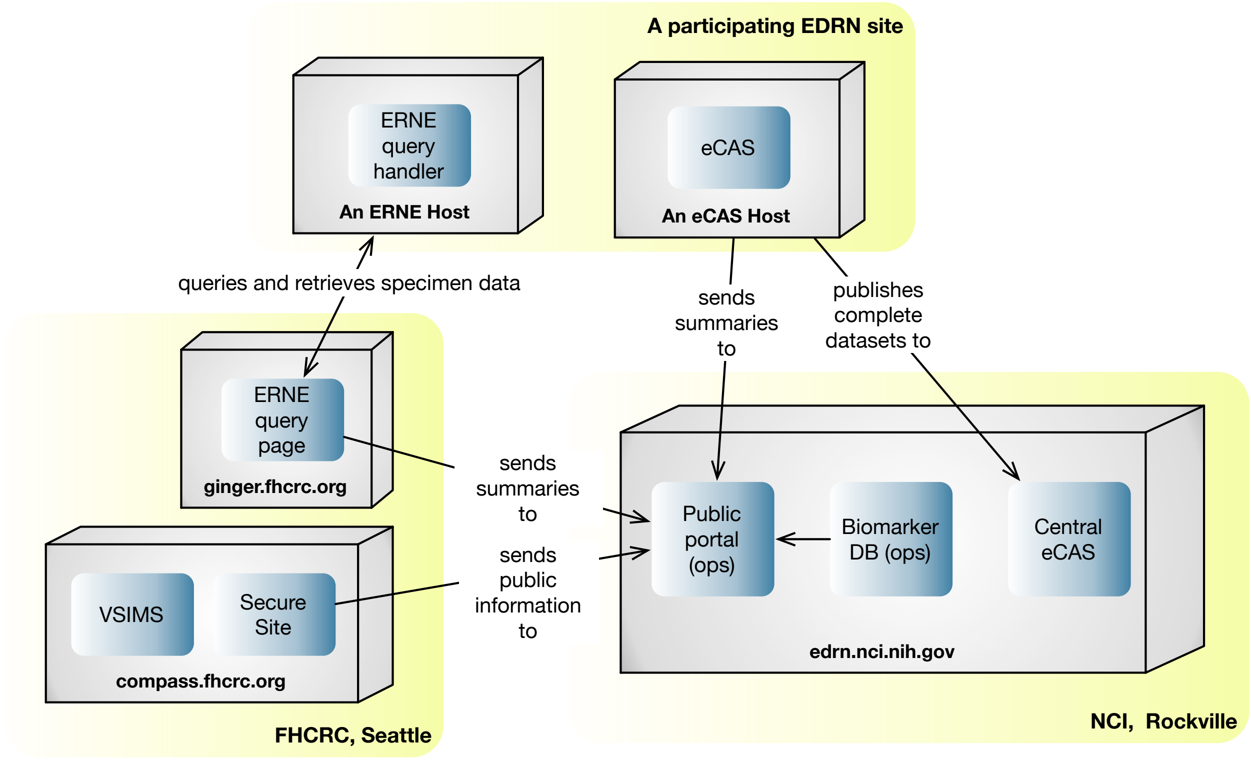
- A single *VSIMS* manages study protocols and related data.

- The *Public Portal* gathers data from all the other applications and presents cross-correlative view of EDRN's knowledge base.

- By publishing standard Application Programmer Interfaces (APIs), EDRN developers (outside of the DMCC and the Informatics Center) may create new *bioinformatics tools* that query the EKE.

## Deployment Model

The EDRN software products are deployed to two central locations and also to many participating EDRN sites. The following diagram depicts the deployment of the products on various hosts:



As shown, FHCRC runs two hosts (ginger.fhcrc.org and compass.fhcrc.org) and NCI runs a single host (edrn.nci.nih.gov). At FHCRC, three applications are installed: the ERNE query web page, VSIMS, and the members-only Secure Site. There are also three applications at NCI: the Public Portal, the Biomarker Database, and the central eCAS.

Each participating EDRN site may then install either (or both, or neither) of an ERNE query handler (queried by the ERNE query web page) and any number of eCAS instances. These services may co-exist on a single host if necessary.

# Release Engineering

The EDRN software products are built from large sets of pre-made components that are glued together with snippets of code, configuration files, and other artifacts to make a complete application. Each application is stored in a source code control repository▸ and each release is tagged so that maintenance of deployed products as well as development of new features may occur simultaneously.

The repository is managed with the Subversion version control system and is hosted at http://oodt.jpl.nasa.gov/repo. Currently, only JPL users may access the repository.

Up until now, there has been an ad hoc release schedule. Software releases were cut in the face of software demands, negotiated with potential users and management as needed.

Both the DMCC and the Informatics Center have resolved to make quarterly releases of software products. That is, four times a year, each of the applications (eCAS, ERNE, VSIMS, etc.) will have a new version released. Applications that have had no new enhancements or bug fixes are exempted from a release in any particular quarter.

## Identifying Releases

Each application will have a numeric release identifier of the following format:

$$x.y.z$$

where *x* is a major release number, *y* is a minor release number, and *z* is a micro release number. The dots are merely separators; no decimal interpretation is intended. The interpretation of the numbers are as follows:

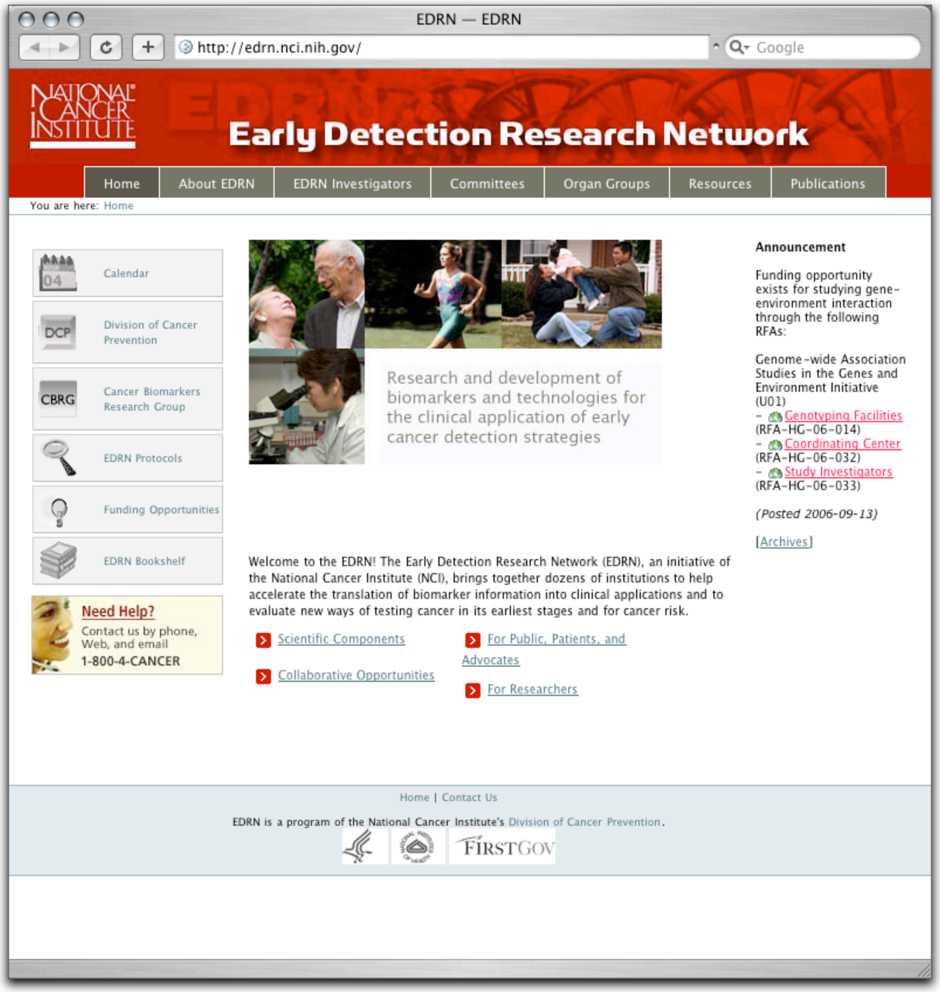
- Bug fixes and enhancements that do not alter existing APIs or function will be marked by increasing *z*, the micro release number.
- Major enhancements that may affect existing APIs or may alter compatibility with prior versions will be marked by increasing *y*, the minor release number.
- Major rewrites that irrevocably alter existing APIs and compatibility will be marked by increasing *x*, the major release number.

Prior to the official cutting of a release, internal identifiers may be used. These take the form of *x.y.z* as above, but also append a lowercase letter, typically sequentially a, b, c, and so forth. No official release may use such designations.

# Public Portal

EDRN's Public Portal is likely the first glimpse into the Early Detection Research Network that most people have. The Public Portal serves as a calling card, brochure, data sheet, white paper, and detailed reference to EDRN's purpose, activities, funding opportunities, and more. It's through the vehicle of the Public Portal that EDRN makes its successes, its collaborative potentials, and indeed its very *raison d'être* known.

Secondly, the Public Portal is a knowledge aggregator. All other EDRN applications publish their data to the Public Portal using the Resource Description Format (RDF), and the Public Portal collates all this data into a searchable index.


EDRN — EDRN
http://edrn.nci.nih.gov/
Google
NATIONAL CANCER INSTITUTE
Early Detection Research Network
Home
About EDRN
EDRN Investigators
Committees
Organ Groups
Resources
Publications
You are here: Home
Calendar
Division of Cancer Prevention
Cancer Biomarkers Research Group
EDRN Protocols
Funding Opportunities
EDRN Bookshelf
Need Help?
Contact us by phone, Web, and email
1-800-4-CANCER
Research and development of biomarkers and technologies for the clinical application of early cancer detection strategies

Welcome to the EDRN! The Early Detection Research Network (EDRN), an initiative of the National Cancer Institute (NCI), brings together dozens of institutions to help accelerate the translation of biomarker information into clinical applications and to evaluate new ways of testing cancer in its earliest stages and for cancer risk.

Scientific Components

For Public, Patients, and Advocates

Collaborative Opportunities

For Researchers

**Announcement**

Funding opportunity exists for studying gene-environment interaction through the following RFAs:

Genome-wide Association Studies in the Genes and Environment Initiative (U01)
- Genotyping Facilities (RFA-HG-06-014)
- Coordinating Center (RFA-HG-06-032)
- Study Investigators (RFA-HG-06-033)

*(Posted 2006-09-13)*

[Archives]

Home | Contact Us

EDRN is a program of the National Cancer Institute's Division of Cancer Prevention.

The portal is located at http://cancer.gov/edrn/. This screen shot is provided for reference.

## Requirements

The requirements of EDRN's Public Portal are as follows:

- The public portal shall be able to link to EDRN web resources that are not hosted on the same platform as the portal.
- The public portal shall share the same data resources used by other EDRN tools including member information, news, and protocol information.
- The public portal shall provide role-based access control for updating by multiple organizations including NCI's Cancer Biomarkers Research Group (CBRG).
- The public portal shall have an uptime of 99.95% including maintenance time.
- The public portal shall support access of up to 20 concurrent users with typical response times being less than 5 seconds.

- The public portal shall provide a publishing mechanism where published news can be reviewed prior to publication.
- The public portal shall support aging of news releases and announcements.
- The pubic portal shall be supported by major web browsers Safari, Firefox, and Opera.
- The public portal shall provide access to information about investigators, on-going studies, meetings, funding opportunities, working groups, scientific discoveries and release of public data sets, publicly available informatics tools, and news.
- Implementation of the public portal shall be done in a way as to enable operation by a separate hosting organization.
- The public portal shall provide single-sign on capability for EDRN Principal Investigators enabling access to other EDRN resources without having to re-login.
- The public portal shall enable access to EDRN informatics tools including ERNE and eCAS.
- The public portal shall be accessible via the URL http://cancer.gov/edrn/.
- The public portal shall have a look and feel similar to other cancer.gov websites.
- The public portal shall have procedures defined for startup, shutdown, and updates as part of a release description document for software releases.
- The public portal shall have a recovery time of two hours in the event of a minor catastrophe, such as minor hardware, network, and facilities problems.
- Basic information on the portal shall be available for update through an administrative interface.
- Extensions for the portal deployment shall be in place to enable new applications and hosting capabilities related to the portal (eCAS, etc).
- Software upgrades shall be available to the developer on a maximum of a week's notice.

## Component Specifications

The Public Portal is built on a hosting platform suitable for dynamic content websites. This section details the components of that platform.

## Hardware Platform

Terrapin Systems LLC‣ hosts and maintains the hardware that runs the Public Portal. While the exact configuration of this platform is unknown, we present the following *recommended* configuration. Should such time that NCI seeks an alternate hosting company or opts to self-host, this table will serve as the specification for the hardware.

http://terpsys.com/

| Specification | Implementation |
|---|---|
| Manufacturer | Apple |
| Model | Xserve |
| Form factor | 1U rack mount |
| CPU | Quad core (2×CPU) 3 GHz Intel Xeon 5100 “Woodcrest” |
| Bus | Independent 1.33 GHz frontside bus per processor |
| Memory | 16 GB (4×4GB) 667 MHz DDR2 ECC FB-DIMMs |
| I/O Bus | PCI Express and PCI-X |
| Network | 2×1000BASE-T (Gigabit) Ethernet onboard |
| Main Storage | 3×300 GB 15000 RPM (software RAID) hot-plug SAS drive modules with SMART |
| Optical storage | DVD+R DL/DVD±RW/CD-RW |
| Other ports | 2×Firewire 800, 1×Firewire 400, 2×USB 2.0, 1×RS232 |
| Power supply | Dual redundant 650 W hot-plug modules |

This hardware platform should be mounted in a 4-post EIA standard rack in an appropriately cooled environment. Access to a major Internet backbone is preferred. The machine supports lights-out remote management and therefore operates “headless” (with no monitor). The operating system (Mac OS X Server) is pre-installed. Operators perform other software installations and management tasks using any combination of

- SSH2
- SNMP
- Apple Remote Desktop

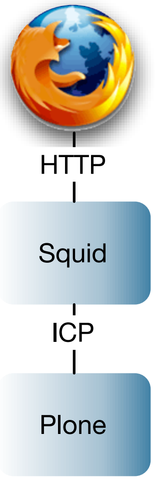
- Apple Server Monitor

## Software Specifications

Support software must be installed on the Public Portal host system prior to installation of any of the EDRN software products. These prerequisite packages are detailed in the following table:

| Package | Version | Source |
|---|---|---|
| Squid | 2.6.4 | http://squid-cache.org/ |
| Python | 2.4.3 | http://python.org/ |
| libjpeg | 6b | http://ijg.org/ |
| FreeType 2 | 2.2.1 | http://freetype.org/ |
| PIL | 1.1.5 | http://www.pythonware.com/products/pil/index.htm |
| Zope | 2.9.5 | http://zope.org/ |
| Plone | 2.5.1 | http://plone.org/ |
| CMFSin | 0.6.1 | http://plone.org/products/cmfsin |

The EDRN public portal runs using the Plone Content Management System (CMS). Plone sits behind the Squid caching web proxy running in its accelerator mode. Browsers connect to squid using HTTP; Squid communicates with Plone using ICP (Internet Caching Protocol).



### Installing the Prerequisite Software

To install the prerequisite software, the following configuration steps are recommended:

1. Install Python by downloading the Mac OS X installer disk image from http://python.org/download/releases/2.4.3/. Open disk image and open the MacPython.mpkg icon.

2. Install Python onto the boot volume. When finished, you may “eject” and discard the disk image.

3. Open Terminal, and create a temporary directory to act as a staging area: `mkdir src; cd src`

4. Add the newly installed Python to the execution PATH:
   - For csh (and compatible) users: `set path = ( /Library/Frameworks/Python.framework/Versions/2.4/bin $path )`
   - For sh (and compatible) users: `PATH=/Library/Frameworks/Python.framework/Versions/2.4/bin:${PATH}`

5. Download and install libjpeg
    - curl 'http://www.ijg.org/files/jpegsrc.v6b.tar.gz' | tar xzf -
    - cd jpeg-6b
    - ./configure; make; sudo make install
    - cd ..; rm -rf jpeg-6b
6. Download and install FreeType 2:
    - curl 'http://download.savannah.gnu.org/releases/freetype/freetype-2.2.1.tar.bz2' | tar xjf -
    - cd freetype-2.2.1
    - ./configure; make; sudo make install
    - cd ..; rm -rf freetype-2.2.1
7. Download and install PIL:
    - curl 'http://effbot.org/downloads/Imaging-1.1.5.tar.gz' | tar xzf -
    - cd Imaging-1.1.5
    - python setup.py build; sudo python.py install
    - cd ..; rm -rf Imaging-1.1.5
8. Download and install Zope:
    - curl 'http://www.zope.org/Products/Zope/2.9.5/Zope-2.9.5-final.tgz' | tar xzf -
    - cd Zope-2.9.5-final
    - ./configure --prefix=/usr/local/zope-2.9.5 --with-python=/Library/Frameworks/Python.framework/Versions/2.4/bin/python
    - make; sudo make install
    - cd ..; rm -rf Zope-2.9.5-final
9. Add Zope to the execution path:
    - For csh (and compatible) users: set path = ( /usr/local/zope-2.9.5/bin $path ); rehash
    - For sh (and compatible) users: PATH=/usr/local/zope-2.9.5/bin:${PATH}
10. Make a new Zope instance: sudo mkzopeinstance.py --dir /usr/local/edrn --user admin:s3cr3t
11. Set the default user for this instance to "nobody":
    - sudo echo zope-user nobody >> /usr/local/edrn/etc/zope.conf
12. Install Squid:

- curl 'http://www.squid-cache.org/Versions/v2/2.6/squid-2.6.STABLE4.tar.bz2' | tar xjf -
- cd squid-2.6.STABLE4
- ./configure; make; sudo make install
- cd ..; rm -rf squid-2.6.STABLE4

13. Configure Squid.
    - Replace the contents of the file /usr/local/etc/squid.conf with http://oodt.jpl.nasa.gov/repo/edrn/portal/trunk/src/conf/squid/squid.conf.
    - Download the following files and place them in /usr/local/etc:
        - http://oodt.jpl.nasa.gov/repo/edrn/portal/trunk/src/conf/squid/iRedirector.py
        - http://oodt.jpl.nasa.gov/repo/edrn/portal/trunk/src/conf/squid/squidAcl.py
        - http://oodt.jpl.nasa.gov/repo/edrn/portal/trunk/src/conf/squid/squidRewriteRules.py
14. Initialize the Squid cache directories:
    - sudo /usr/local/sbin/squid -z
15. Place Plone products into the Zope instance:
    - cd /usr/local/edrn/Products
    - curl 'http://easynews.dl.sourceforge.net/sourceforge/plone/Plone-2.5.1-final.tar.gz' | sudo tar xzf -
    - cd Plone-2.5.1
    - sudo mv * ..
    - cd ..; sudo rm -rf Plone-2.5.1
16. Place the CacheFu product into the Zope instance:
    - cd /usr/local/edrn/Products/CacheFu
    - sudo mv CacheSetup PageCacheManager CMFSquidTool PolicyHTTPCacheManager ..
17. Place the CMFSin product into the Zope instance:
    - cd /usr/local/edrn/Products
    - curl 'http://plone.org/products/cmfsin/releases/0.6.1/CMFSin-0.6.1.tar.gz' | sudo tar xzf -
18. Install launchd service initialization files by placing the two files referenced below into /Library/LaunchDaemons:

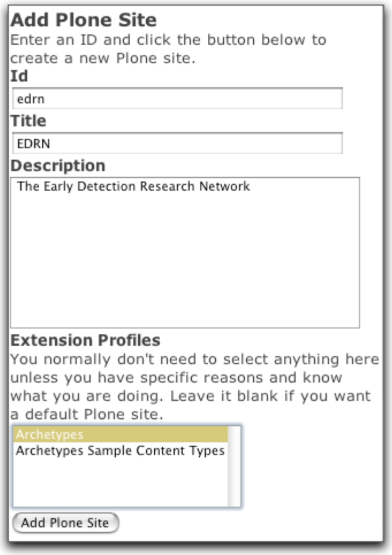
- http://oodt.jpl.nasa.gov/repo/edrn/portal/trunk/src/conf/LaunchDaemons/EDRNSquid.plist
- http://oodt.jpl.nasa.gov/repo/edrn/portal/trunk/src/conf/LaunchDaemons/EDRNZope.plist

19. Start Zope:
    - sudo launchctl start gov.nih.nci.edrn.Zope

20. Using a browser, visit the Zope Management Interface (ZMI) at http://localhost:8080/manage logging in with user name admin and password s3cr3t.

21. From the ZMI, and add a new Plone Site object with the following parameters:

| Parameter | Setting |
|---|---|
| Id | edrn |
| Title | EDRN |
| Description | The Early Detection Research Network. |
| Extension Profiles | Archetypes |



**Add Plone Site**

Enter an ID and click the button below to create a new Plone site.

**Id**

edrn

**Title**

EDRN

**Description**

The Early Detection Research Network

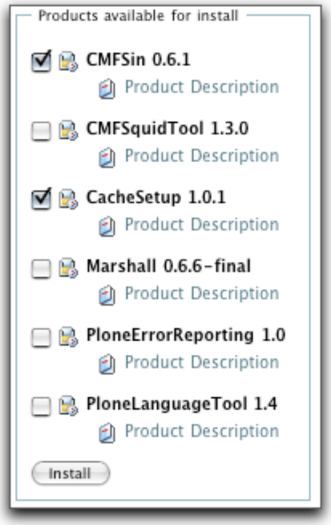
**Extension Profiles**

You normally don't need to select anything here unless you have specific reasons and know what you are doing. Leave it blank if you want a default Plone site.

Archetypes

Archetypes Sample Content Types

22. Visit the Plone site at http://localhost:8080/edrn/ and install CacheFu and CMFSin:
    - Click Site Setup
    - Click Add/Remove Products
    - Check the box next to CMFSin 0.6.1
    - Check the box next to CacheSetup 1.0.1
    - Click the Install button



Products available for install

☑ CMFSin 0.6.1
Product Description

□ CMFSquidTool 1.3.0
Product Description

☑ CacheSetup 1.0.1
Product Description

□ Marshall 0.6.6-final
Product Description

□ PloneErrorReporting 1.0
Product Description

□ PloneLanguageTool 1.4
Product Description

23. Start Squid:
    - sudo launchctl start gov.nih.nci.edrn.Squid

24. Change the admin password in the ZMI to a secure password that only you know.

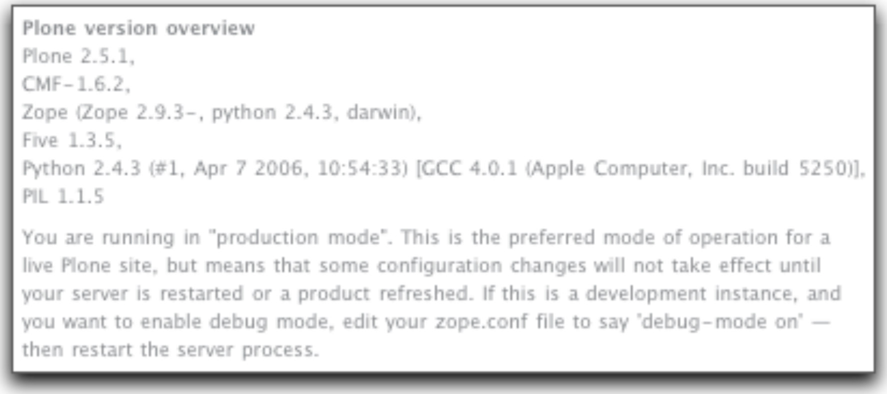
The prerequisite software should now be fully installed and ready for use. The next steps would be to test to ensure all prerequisites are functioning properly, and then to install the EDRN Public Portal products.

### Testing the Prerequisite Software

Since there are many software components on which the Public Portal depends, the installation must be tested before proceeding to install the EDRN Public Portal products. This section provides procedures for testing the installation. If the installation procedure described above was followed rigorously, then these tests should pass easily.

Follow these steps in order to test the prerequisite software:

1. Reboot the host computer.
2. Using a browser an a system external to the host, visit `http://edrn.nci.nih.gov/`. If the Plone default page fails to appear:
   - Check that Squid is running. If not, check the launchd log as well as the file `/var/log/squid/squid.log` to determine why it failed to start.
   - Check that Zope is running. If not, check the launchd log as well as the file `/usr/local/edrn/log/event.log` to determine why it failed to start.
3. Click the log in link and log in using user name admin and the new administrator password you set during installation.
4. Click the "Site Setup" link. Ensure the bottom of the page under "Plone version overview" appears similarly to the following:



**Plone version overview**

```
Plone 2.5.1,
CMF-1.6.2,
Zope (Zope 2.9.3-, python 2.4.3, darwin),
Five 1.3.5,
Python 2.4.3 (#1, Apr 7 2006, 10:54:33) [GCC 4.0.1 (Apple Computer, Inc. build 5250)],
PIL 1.1.5
```

You are running in "production mode". This is the preferred mode of operation for a live Plone site, but means that some configuration changes will not take effect until your server is restarted or a product refreshed. If this is a development instance, and you want to enable debug mode, edit your zope.conf file to say 'debug-mode on' — then restart the server process.

If these steps all pass, you're now ready to install the EDRN Public Portal software products.

## Installing the Public Portal Products

With a working Plone installation, you can install the products that comprise the EDRN public portal. These products are provided by the Informatics Center as Zope products and are installed as other Zope products (by extracting them into the Products directory of the Zope instance).

There are three products that must be installed:

- EDRNPortalSkin. This product sets up the look and feel for the EDRN Public Portal, similar to that of other cancer.gov websites.

- ERNE. This product provides an interface to the Virtual Specimen Bank from within the portal.
- EKE. This product provides the searchable index of all EDRN knowledge that other EDRN applications populate via RDF.

## Installing the Products

To install the products, you check them out of the Informatics Center's Subversion repository and into your /usr/local/edrn/Products directory. After starting the Zope server, you can then visit Plone's "Add/Remove Products" page to install each one.

The following steps detail exactly what to do:

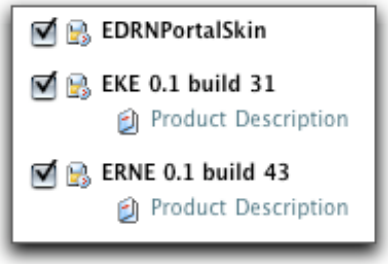
1. Shut down the Zope instance server:

   -
     ```
     sudo launchctl stop gov.nih.nci.edrn.Zope
     ```
2. Go to the instance's Product directory:

   -
     ```
     cd /usr/local/edrn/Products
     ```
3. Check out the products:

   -
     ```
     sudo svn checkout http://oodt.jpl.nasa.gov/repo/eke/trunk/src/plone/{EDRNPortalSkin,ERNE,EKE}
     ```
4. Start the Zope instance server:

   -
     ```
     sudo launchctl start gov.nih.nci.edrn.Zope
     ```
5. Visit the Site Setup at http://edrn.nci.nih.gov/plone_control_panel
   - Log in when prompted
   - Click Add/Remove Products
   - Check the boxes next to EDRNPortalSkin, ERNE, and EKE.
   - Click the Install button



☑ EDRNPortalSkin

☑ EKE 0.1 build 31
Product Description

☑ ERNE 0.1 build 43
Product Description

## Testing the Installation

To test if the EDRN software products that implement the Public Portal were successfully installed, perform the tests described in this section. The tests involve checking each of the products.

### Checking the Portal Skin

The EDRNPortalSkin product provides the cancer.gov look and feel for the EDRN Public Portal. While it's subjective to determine if the Public Portal happens to look like other cancer.gov websites, there are definite features of the portal that can be checked.

To perform these checks, perform the following steps:

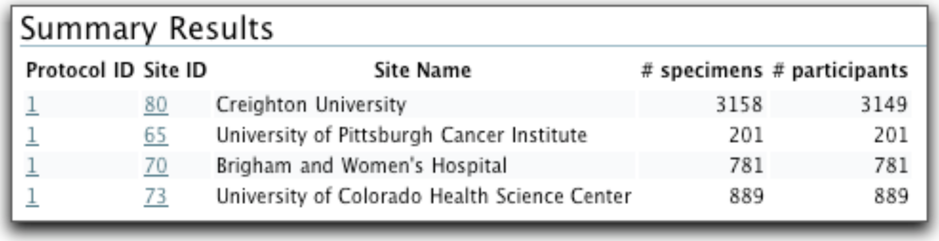
1. Visit http://edrn.nci.nih.gov/ with a browser
2. Confirm that the right-side of the page contains at least one article. If not, confirm that the DMCC is providing RSS feeds.
3. Click the Archives link under the right-side article.
4. Confirm that there is a history of articles. If not, confirm that the DMCC is providing RSS feeds.

### Checking the ERNE Application

The ERNE product provides a a way to query a simulated ERNE deployment from within the Public Portal. It does so by populated the portal with simulated specimen records.

To test the ERNE Application, perform the following steps:

1. Visit http://edrn.nci.nih.gov/erne-search/ with a browser.
2. Click the Submit button at the bottom of the page.
3. Ensure that specimen results return from at least four sites. Note that response may vary based on data population. If no results are returned, check the server's log in /usr/local/edrn/log/event.log to diagnose the problem; then repair the problem.



Summary Results

| Protocol ID | Site ID | Site Name | # specimens | # participants |
|---|---|---|---|---|
| 1 | 80 | Creighton University | 3158 | 3149 |
| 1 | 65 | University of Pittsburgh Cancer Institute | 201 | 201 |
| 1 | 70 | Brigham and Women's Hospital | 781 | 781 |
| 1 | 73 | University of Colorado Health Science Center | 889 | 889 |

### Checking the EDRN Knowledge Environment

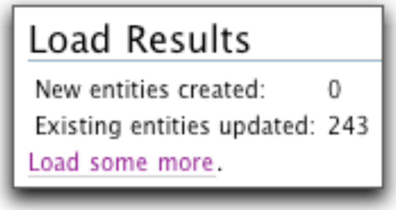
The EKE product provides for the unified EDRN Knowledge Environment, an application that unifies EDRN's data artifacts into a correlated knowledge-base. This lets researchers locate interesting data and find relationships they may not have realized existed, speeding scientific discovery.

To test the EDRN Knowledge Environment, it must first be loaded with entity data. The central EDRN Catalog and Archive System is one such source of data. Perform the following steps to test the EDRN Knowledge Environment:

1. Visit http://edrn.nci.nih.gov/load/ with a browser.

2. In the URL field, enter http://edrn.nci.nih.gov/filemgr/rdf?type=ALL and press the Submit button.

3. Ensure that either the “New entities created” or “Existing entities updated” (or both) are greater than zero. If both are zero, check that the central EDRN Catalog and Archive System is running properly. If an error message appears, use it to diagnose and repair the problem.



Load Results

New entities created: 0
Existing entities updated: 243
Load some more.

4. Visit http://edrn.nci.nih.gov/search/index and enter “cancer” as a search term.

5. Ensure that more than one match appears. If no matches appear, check the file /usr/local/edrn/log/event.log for diagnostic messages and repair the problem.

## Regular Maintenance

The EDRN Public Portal must undergo regular maintenance that ensures that, should a disastrous tragedy befall, the system could be restored rapidly. This section describes the routine maintenance that must occur.

### Backups

Routine system dumps must be performed on the Public Portal software in order to maintain changes to the content which may occur daily (though so far such updates are far less frequent). While regular system dumps are certainly a necessary task for the hosting hardware, they are not sufficient for the software that supports the Public Portal.

There are four artifacts that must be backed up:

- The file /usr/local/edrn/etc/zope.conf
- The directory /usr/local/edrn/Products
- Any customizations to internationalization files that may have been made in /usr/local/edrn/var/i18n
- The Zope Object Database

Of the above, the first three may be backed up using standard system dump utilities. However, the Zope Object Database requires a specialized utility to back up. The artifacts produced by this utility may then be backed up with the standard system dump utilities.

To back up the Zope Object Database, we use the Repozo utility. A backup script is available at:

- `http://oodt.jpl.nasa.gov/repo/edrn/portal/trunk/src/conf/LaunchDaemons/PortalBackup.plist`

Install this file in /Library/LaunchDaemons. This will enable the system to begin full and incremental backups into /usr/local/edrn/var/backups automatically. The contents of this directory may be backed up in the usual way with the system dump utility.

For that directory and the /usr/local/edrn/etc/zope.conf file, the /usr/local/edrn/Products directory, and the /usr/local/edrn/var/i18n directory, we recommend the following backup schedule:

- On Sunday of each week, run a full dump.
- Daily on Monday through Saturday, run an incremental dump.
- Once a year, remove a copy of the full dump to an off-site location. We recommend a safety deposit box at a Swiss bank which operates a branch office in Paris.

# Study Information Management

requirements, installation, operation, test, support guide here

# Catalog and Archive Service

requirements, installation, operation, test, support guide here

# Biomarker Database

requirements, installation, operation, test, support guide here

# Virtual Specimen Bank

**ERNE here.**

# Secure Site

secure site. VSIMS, secure web site, ERNE, and we need LDAP!

# Appendix